

### **Remarks**

Reconsideration of the application is respectfully requested in view of the foregoing amendments and following remarks. Claims 1-23, 26-32, 34-37, 39-41, and 43 are pending in the application. No claims have been allowed. Claims 1, 15, 31, 34, 39, and 43 are independent. Claims 1, 7, and 15 have been amended. Claim 16 is canceled.

### **Cited Art**

The Office action ("Action") applies the following cited art: U.S. Patent Number 6,560,774 to Gordon *et al.* ("Gordon.")

### **Initialed Form 1449 Not Received**

On May 3, 2006, Applicants submitted an Information Disclosure Statement listing several patent references and two non-patent references. On December 28, 2007, Applicants submitted an Information Disclosure Statement listing one reference. On January 29, 2008, Applicants submitted an Information Disclosure Statement listing one reference. The patent references for the May 3, 2006, IDS were initialed, but the non-patent references were not. Applicants have not yet received initialed 1449 forms for the other IDS submissions. Applicants respectfully request that the Examiner provide the initialed 1449 forms for these IDS submissions. See MPEP § 609 ("An information disclosure statement filed in accordance with the provisions of 37 CFR 1.97 and 37 CFR 1.98 will be considered by the examiner assigned to the application.").

### **§ 102 Rejections**

The Action rejected claims 1-15, 17-23, 26-32, 34-37, 39-41, and 43 under 35 U.S.C. § 102(a) as anticipated by Gordon. Applicant respectfully submits the claims are allowable over the cited art. The rejections are traversed.

**Claims 1-14 are Allowable Over Gordon**

Claim 1 recites one or more computer-readable media with computer-executable instructions for implementing a software development architecture comprising, in part:

a code generator operable to generate code targeted for a plurality of execution architectures;

wherein the code generator constructs components of software development tools using the software development scenario-independent intermediate representation format, the one or more exception handling models, and the type system.

Gordon does not teach or suggest the above recited language of claim 1.

Gordon states that a problem in the art is that, "[a] difficulty arises in intermediate language-type models in that, in some circumstances, the execution engine needs to run untrusted code." (Gordon, 24-26.) As a proposed solution to this problem, Gordon describes "a verifier to check intermediate language code." (Gordon, 1:45-46.) Gordon states that:

In one embodiment, a computer-implemented method first verifies metadata of an intermediate language code for consistency and accuracy, and then verifying the intermediate language code for consistency and accuracy. This latter part in one embodiment is accomplished by performing first a syntactic check of the intermediate language code, and then a semantic check of the intermediate language code. (Gordon, 1:46-53.)

Thus, Gordon describes a code verification method that verifies code in several different ways to ensure consistency and accuracy of untrusted code. At no point does Gordon teach or suggest software development tools, much less the above language of claim 1 reciting "components of software development tools [constructed] using the software development scenario-independent intermediate representation format, the one or more exception handling models, and the type system."

The Action alleges, for example with respect to claim 31 and elsewhere, that Gordon describes creating software development tools at, for example, Figure 23 and related text referring to JIT compilers. (Action at, *e.g.*, page 10.) Applicants respectfully disagree.

Figure 23 of Gordon shows a system flow of a "Verifier" going into a system module comprising "JIT compilers (EconoJIT, JIT, OptJIT, or Third-Party JIT)." The related text of Gordon states:

The Execution Engine creates an environment for code execution called the Virtual Execution System, which is shown in FIG. 23. In most cases, source code is compiled into IL, **and the IL is loaded and executed on-the-fly using one of the JIT compilers to convert the IL to native code.**

(Gordon, 27:12-16, emphasis added by Applicants.

Thus, Gordon is not constructing software development tools, or components of software development tools as recited in claim 1, but is loading intermediate language code and executing the intermediate language code using a JIT compiler to convert the intermediate language code to native code. Gordon does not teach or suggest the above recited language of claim 1.

For at least these reasons, Gordon does not teach or suggest the language of claim 1 or its dependant claims 2-14. Applicants respectfully request withdrawal of the § 102(a) rejection and allowance of claim 1 and its dependant claims 2-14.

**Claims 15, 17-23, and 26-30 are Allowable over Gordon**

Claim 15 recites a method of creating a target software development tool, the method comprising in part:

compiling the at least one software development component and framework to create the target software development tool.

Gordon does not teach or suggest the above recited language of independent claim 15. As discussed above, Gordon describes a code verification method that verifies code in several different ways to ensure consistency and accuracy of untrusted code. At no point does Gordon teach or suggest software development tools, much less the language of independent claim 15 reciting "compiling the at least one software development component and framework to create the target software development tool."

Additionally, independent 15 has been amended to incorporate the language of canceled claim 16. The Action alleges with respect to claim 16 that Gordon describes creating software development tools at, for example, Figure 23 and related text referring to JIT compilers. (Action, page 7.) Applicants respectfully disagree. As discussed above, Gordon describes loading intermediate language

code and executing the intermediate language code using a JIT compiler to convert the intermediate language code to native code. Thus, Gordon clearly does not teach or suggest "compiling the at least one software development component and framework to create the target software development tool," as recited by independent claim 15.

For at least these reasons, Gordon does not teach or suggest the above recited language of independent claim 15 or its dependant claims 17-23 and 26-30. Applicants respectfully request withdrawal of the § 102(a) rejection and allowance of independent claim 15 and its dependant claims 17-23 and 26-30.

### **Claims 31 and 32 are Allowable over Gordon**

Claim 31 recites a method of creating a target software development tool from a common framework, the method comprising in part:

creating the target software development tool from the integrated common framework.

Gordon does not teach or suggest the above recited language of independent claim 31. As discussed above, Gordon describes a code verification method that verifies code in several different ways to ensure consistency and accuracy of untrusted code. At no point does Gordon teach or suggest software development tools, much less the language of claim 31 reciting "creating the target software development tool from the integrated common framework."

Additionally, the Action alleges with respect to claim 31 that Gordon describes creating software development tools at, for example, Figure 23 and related text referring to JIT compilers. (Action, page 10.) Applicants respectfully disagree. As discussed above, Gordon describes loading intermediate language code and executing the intermediate language code using a JIT compiler to convert the intermediate language code to native code. Thus, Gordon clearly does not teach or suggest "creating the target software development tool from the integrated common framework," as recited by independent claim 31.

For at least these reasons, Gordon does not teach or suggest the above recited language of independent claim 31 or its dependant claim 32. Applicants respectfully request withdrawal of the § 102(a) rejection and allowance of independent claim 31 and its dependant claim 32.

**Claims 34-37 are Allowable over Gordon**

Claim 34 recites a method of producing inter-compatible software development tools, the method comprising in part:

creating a first software development tool from a software development architecture that is operable to support a plurality of different programming languages.

Gordon does not teach or suggest the above recited language of independent claim 34. As discussed above, Gordon describes a code verification method that verifies code in several different ways to ensure consistency and accuracy of untrusted code. At no point does Gordon teach or suggest software development tools, much less the language of claim 34 reciting "creating a first software development tool from a software development architecture."

Additionally, the Action alleges with respect to claim 34 that Gordon describes creating software development tools at, for example, Figure 23 and related text referring to JIT compilers. (Action, page 11.) Applicants respectfully disagree. As discussed above, Gordon describes loading intermediate language code and executing the intermediate language code using a JIT compiler to convert the intermediate language code to native code. Thus, Gordon clearly does not teach or suggest "creating a first software development tool from a software development architecture that is operable to support a plurality of different programming languages," as recited by independent claim 34.

For at least these reasons, Gordon does not teach or suggest the above recited language of independent claim 34 or its dependant claims 35-37. Applicants respectfully request withdrawal of the § 102(a) rejection and allowance of independent claim 34 and its dependant claims 35-37.

**Claims 39-41 are Allowable over Gordon**

Claim 39 recites a method of modifying a software development tool, the software development tool having been created using a software development architecture that is operable for a plurality of different programming languages and comprising one or more software development components, the method comprising in part:

creating a modified software development tool from the dynamically linked binary version and the software development component.

Gordon does not teach or suggest the above recited language of independent claim 39. As discussed above, Gordon describes a code verification method that verifies code in several different

ways to ensure consistency and accuracy of untrusted code. At no point does Gordon teach or suggest software development tools, much less the language of claim 39 reciting "creating a modified software development tool from the dynamically linked binary version and the software development component."

Additionally, the Action alleges with respect to claim 39 that Gordon describes creating software development tools at, for example, Figure 23 and related text referring to a linker. (Action, page 13.) Applicants respectfully disagree. As discussed above, Gordon describes loading intermediate language code and executing the intermediate language code using a JIT compiler to convert the intermediate language code to native code. Thus, Gordon clearly does not teach or suggest "creating a modified software development tool from the dynamically linked binary version and the software development component," as recited by independent claim 39.

For at least these reasons, Gordon does not teach or suggest the above recited language of independent claim 39 or its dependant claims 40 and 41. Applicants respectfully request withdrawal of the § 102(a) rejection and allowance of independent claim 39 and its dependant claims 40 and 41.

**Claim 43 is Allowable over Gordon**

Claim 43 recites a method of creating a software development tool, the method comprising in part:

creating the software development tool via the linked software component and computer-executable file.

Gordon does not teach or suggest the above recited language of independent claim 43. As discussed above, Gordon describes a code verification method that verifies code in several different ways to ensure consistency and accuracy of untrusted code. At no point does Gordon teach or suggest software development tools, much less the language of claim 43 reciting "creating the software development tool via the linked software component and computer-executable file."

Additionally, the Action alleges with respect to claim 43 that Gordon describes creating software development tools at, for example, Figure 2 and related text. Applicants respectfully disagree. Gordon describes:

Multiple source code language sources . . . are compiled, by compilers such as compilers 204, into intermediate language (IL) code. The execution engine 200 then compiles, interprets, or just-in-time compiles the IL code into executable code . . . The verifier 202 verifies the IL code prior to the code being compiled, interpreted, or just-in-time compiled . . .

In one embodiment, the verifier 202 verifies the intermediate language code as well as metadata of the IL code . . .  
(Gordon, 6:11-23.)

Gordon thus described compiling multiple source languages into IL code which is input to a verifier for verification. At no point, however, does Gordon teach or suggest a software development tool, much less "creating the software development tool via the linked software component and computer-executable file."

Thus, Gordon clearly does not teach or suggest "creating the software development tool via the linked software component and computer-executable file," as recited by independent claim 43.

For at least these reasons, Gordon does not teach or suggest the above recited language of independent claim 43. Applicants respectfully request withdrawal of the § 102(a) rejection and allowance of independent claim 43.

### **Request for Interview**

If any issues remain, the Examiner is formally requested to contact the undersigned attorney in order to arrange a telephonic interview. It is believed that a brief discussion of the merits of the present application may expedite prosecution. Applicants submit the foregoing formal Amendment so that the Examiner may fully evaluate Applicants' position, thereby enabling the interview to be more focused.

This request is being submitted under MPEP § 713.01, which indicates that an interview may be arranged in advance by a written request.

### **Conclusion**

The claims should be allowable. Such action is respectfully requested.

Respectfully submitted,

KLARQUIST SPARKMAN, LLP

One World Trade Center, Suite 1600  
121 S.W. Salmon Street  
Portland, Oregon 97204  
Telephone: (503) 595-5300  
Facsimile: (503) 595-5301

By /Cory A. Jones/  
Cory A. Jones  
Registration No. 55,307